94

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/085,455	02/27/2002	Motohiro Kawahito	JP920000420US1	1801

7590 07/26/2006

ANNE V. DOUGHERTY
3173 CEDAR RD.
YORKTOWN HEIGHTS, NY 10598

EXAMINER
PHAM, CHRYSTINE

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 07/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/085,455	**Applicant(s)** KAWAHITO ET AL.	
	Examiner Chrystine Pham	**Art Unit** 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 April 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to Paper filed on April 13, 2006. No claims have been amended. Claims 1-16 are pending.

Response to Arguments

2. Applicant's arguments filed April 13, 2006 have been fully considered but they are not persuasive.

Essentially, Applicants contend, "Linden does not teach dynamically generating a path along which a **parameter** for the predetermined command is **fixed in a specific state**" (Emphasis added)(Remarks, page 10, last paragraph). The Examiner respectfully disagrees.

First, in paragraph [0034] (associated with FIG.2), Linden explicitly discloses the dynamic compiler decoding (i.e., determining) the source instructions and their parameters as a first step. The same passage explicitly discloses the **dynamic** compiler **creating/generating** an instruction stream (i.e., **path**) that is optimized **based on** said decoded (i.e., **predetermined**) **source instructions and parameters**. As established in previous Office Action, paragraphs [0037]-[0038] and [0041] of Linden explicitly discloses the decoding stage going through block of source instructions and analyzes the operation codes such as the addition (i.e., command) of two numbers (i.e., parameters). Specifically in paragraph [0038], Linden explicitly discloses translating the assignment (i.e., command) of Register 3 (i.e., one of the parameters) to the outcome of "Register 1 XOR

Register 1" which is always fixed in a specific state because clearly it does not matter what the value of variable Register 1 is, the result of XOR-ing a variable and itself is always the constant 0 (i.e., Boolean false). In response to Applicants' argument that Linden creates new instructions for the target device "*independent of operations specified by the source instructions*", it is submitted that the Applicants quoting of Linden is out of context because in paragraph [0041] (last 7 lines), Linden explicitly states that:

"***One can look at this optimization process*** ***as*** *an interpolation from the source instructions to the equivalent results to be achieved by the target processor, independent of the operations specified by the source instructions, but* ***dependent on the intended purpose and flow of the source instructions and how they are to be handled by the target processor to achieve an equivalent result.***"

Thus, in this passage, Linden is merely offering a "bird's-eye" view for looking at the optimization process, which in fact is dependent on the intended purpose of the source instructions and the instruction stream generated from said source instructions during the decoding stage. Without considering the source instructions (and associated parameters), it is impossible to derive (i.e., generate) equivalent and optimized instructions to be executed by the target processor.

Applicants similarly argue that Linden does not teach "detecting one command, ... for which a variable can be limited to a predetermined constant value, and for which the processing speed can be increased" (Remarks, page 13, first full paragraph). As discussed above, paragraphs [0037]-[0038] of Linden clearly discloses optimizing the source program by generating an new instruction for the source instruction where the result is always a constant, for example by eliminating certain operations (e.g., Register 1 XOR Register 1). Needless to say, optimization by removing parameters and/or operations clearly anticipates increasing the processing speed since less parameters and operations would have to be processed in the target processor.

3. In view of the fore going discussion, rejection of the claims under USC 102(e) and 103(a) is considered proper and maintained.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the

United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-4, 6-8, 10-16 are rejected under 35 U.S.C. 102(e) as being anticipated by Linden, (US 2002/0066086 A1), hereinafter, *Linden.*

Claim 1

Linden teaches a program optimization method (see at least *optimizing 14* FIG.2 & associated text; *14, 16* FIG.3 & associated text) for translating, into machine code (see at least *target instructions executable* paragraph [0003]), source code for a program written in a programming language (see at least *source instructions* paragraph [0003]), and for dynamically optimizing said program (see at least *dynamic compiler 10*, *optimizing 14* FIG.2 & associated text) comprising the steps of:

- performing a dynamic analysis during execution to determine whether the execution speed of said program can be increased by fixing, in a specific state, a parameter for a predetermined command in said program (see at least Abstract; *dynamically cross-compiling, execution speed, execution time, overhead* paragraphs [0010]-[0012]; *dynamic recompilation* paragraphs [0015]-[0016]; *decoded instruction, instruction sequence, result, constant, Register 3=0* paragraphs [0038], [0049]); and
- employing results of said analysis for the dynamic generation, in said program, of a path along which said parameter of said predetermined command is fixed in said specific state (see at least *instruction sequence, result, constant, Register*

3=0,optimization step 44, optimized instruction stream paragraph [0038]; see at least *decoding stage, optimization stage, flow of information, encoding stage* paragraph [0018]; paragraph [0037]; paragraph [0041]).

Claim 2

The rejection of base claim 1 is incorporated. *Linden* further teaches wherein said step of generating a path includes the steps of:

- executing said program and obtaining statistical data for the appearance frequency of each available state (see at least *cross-compilation, execution loops of instructions* paragraph [0012]; paragraphs [0038]-[0039]) wherein, according to said results of said analysis, said parameter of said predetermined command may be set (see at least paragraphs [0038]-[0039]); and
- employing said obtained statistical data to dynamically generate said path (see at least *instruction sequence, result, constant, Register 3=0,optimization step 44, optimized instruction stream* paragraph [0038]; see at least *decoding stage, optimization stage, flow of information, encoding stage* paragraph [0018]; paragraph [0037]; paragraph [0041]).

Claim 3

Linden teaches a program optimization method (see at least *optimizing 14* FIG.2 & associated text; *14, 16* FIG.3 & associated text), the source code for a program written in a programming language (see at least *source instructions* paragraph [0003]), and for optimizing said program comprising the steps of:

- executing a program to obtain statistical data for an appearance frequency of each available state in which a parameter of a predetermined command in said program may be set (see at least *cross-compilation, execution loops of instructions* paragraph [0012]; paragraphs [0038]-[0039]); and
- employing said obtained statistical data to dynamically generate a machine language program that includes, as the compiling results, a path (see at least *instruction sequence, result, constant, Register 3=0,optimization step 44, optimized instruction stream* paragraph [0038]; see at least *decoding stage, optimization stage, flow of information, encoding stage* paragraph [0018]; paragraph [0037]; paragraph [0041]).

Claim 4

The rejection of base claim 3 is incorporated. *Linden* further teaches comprising a step of: generating a machine language program that does not include, as a compiling result, a path along which said parameter of said predetermined command is fixed in a specific state (see at least *instruction sequence, result, constant* paragraph [0038]).

Claim 6

Linden teaches a program optimization method for translating, into machine code, the source code for a program written in a programming language, and for optimizing said program comprising the steps of:

- detecting dynamically during program execution one command, of the commands in said program, for which a variable can be limited to a predetermined constant value, and for which the processing speed can be increased by limiting said variable to said constant value (see at least *instruction sequence, result, constant, Register 3=0* paragraph [0038]); and
- generating a path along which said constant value of said variable of said detected command is fixed (see at least *instruction sequence, result, constant, Register 3=0,optimization step 44, optimized instruction stream* paragraph [0038]).

Claim 7

Linden teaches a dynamic compiler (see at least *dynamic compiler 10*, *optimizing 14* FIG.2 & associated text) for translating into machine code the source code for a program written in a programming language (see at least *source instructions, target instructions executable* paragraph [0003]), and for optimizing the resultant program (see at least *dynamic compiler 10*, *optimizing 14* FIG.2 & associated text) comprising:

- an impact analysis unit for performing an analysis to dynamically determine during execution how much the execution speed of said program can be increased by fixing, in a specific state, a parameter of a predetermined command in said program (see at least Abstract; *dynamically cross-compiling, execution speed, execution time, overhead* paragraphs [0010]-[0012]; *dynamic recompilation* paragraphs [0015]-[0016]; *decoded instruction, instruction sequence, result, constant, Register 3=0* paragraphs [0038], [0049]); and

- a specialization unit for employing the analysis results obtained by said impact analysis unit to generate, in said program, a specialized path along which said parameter of said predetermined command is fixed in said specific state (see at least *instruction sequence, result, constant, Register 3=0,optimization step 44, optimized instruction stream* paragraph [0038]).

Claim 8

The rejection of base claim 7 is incorporated. *Linden* further teaches:

- a data specialization selector for, when said program is executed, obtaining statistical data for the appearance frequency of each state obtained by said impact analysis unit, and for determining the state in which said parameter of said predetermined command is to be set (see at least *cross-compilation, execution loops of instructions* paragraph [0012]; paragraphs [0038]-[0039]),
- wherein said specialization unit generates a specialized path along which said parameter of said predetermined command is fixed in a state determined by said data specialization selector (see at least *instruction sequence, result, constant, Register 3=0,optimization step 44, optimized instruction stream* paragraph [0038]; see at least *decoding stage, optimization stage, flow of*

information, encoding stage paragraph [0018]; paragraph [0037]; paragraph [0041]).

Claim 9

The rejection of base claim 8 is incorporated. *Linden* further teaches wherein, in accordance with the state of said program at execution, said specialization unit generates, in said program, a branching process for selectively performing a specialized path and an unspecialized path; and wherein, while taking into account a delay due to the insertion of said branching process, said data specialization selector determines a state in which said parameter of said predetermined command is fixed (see at least *optimized instruction flow stream, optimization rules, pipeline delay* paragraphs [0046]-[0047]).

Claim 10

Linden teaches a computer (see at least FIG.1 & associated text) comprising:

- an input device for receiving source code for a program (see at least *42* Fig.3 & associated text);
- a dynamic compiler (see at least *10* FIG.2 & associated text) for translating said source code to compile said program and for converting said compiled program into machine language code (see at least *source instructions, target instructions executable* paragraph [0003]); and

- a processor for executing said machine language code (see at least *22* FIG.1 & associated text),
- wherein said dynamic compiler includes
- means for performing an dynamic analysis to determine during execution whether the execution speed of said program can be improved by fixing in a specific state a parameter of a predetermined command in said program (see at least Abstract; *dynamically cross-compiling, execution speed, execution time, overhead* paragraphs [0010]-[0012]; *dynamic recompilation* paragraphs [0015]-[0016]; *decoded instruction, instruction sequence, result, constant, Register 3=0* paragraphs [0038], [0049]), and
- means for generating in said program, based on the analysis results, a path along which said parameter of said predetermined command is fixed in said specific state and for compiling said program (see at least *instruction sequence, result, constant, Register 3=0,optimization step 44, optimized instruction stream* paragraph [0038]), and
- wherein said compiler outputs, as the compiled results, said machine language code that includes said path along which the state of said parameter is fixed (see at least *instruction sequence, result, constant, Register 3=0,optimization step 44, optimized instruction stream* paragraph [0038]; see at least *decoding stage, optimization stage, flow of information, encoding stage* paragraph [0018]; paragraph [0037]; paragraph [0041]).

Claim 11

Linden teaches a computer comprising:

- an input device, for receiving source code for a program (see at least *42* Fig.3 & associated text);
- a dynamic compiler (see at least *10* FIG.2 & associated text), for translating said source code to compile said program and for converting said compiled program into machine language cod (see at least *source instructions, target instructions executable* paragraph [0003])e; and
- a processor, for executing said machine language code (see at least *22* FIG.1 & associated text),
- wherein said dynamic compiler includes
- means for obtaining statistical data for the appearance frequency of each available state wherein a parameter for a predetermined command in said program may be set when said program is executed, and for employing said statistical data to determine a state in which said parameter of said predetermined command is to be fixed (e.g., see *inductive expressions, multiplications, additions* col.4:1-10), and
- means for generating a specialized path along which said parameter of said predetermined command is fixed in said determined state, and for compiling said program (e.g., see *constant expression evaluation routine, runtime, object code image, Kfold routine* col.22:6-20; see *generating code* col.22:67-col.23:8; see *machine code, constant expression evaluation routine* col.23:32-35), and

- wherein said compiler outputs, as the compiled results, said program as said machine language code that includes said specialized path (e.g., see *constant expression evaluation routine, runtime, object code image, Kfold routine* col.22:6-20; see *generating code* col.22:67-col.23:8; see *machine code, constant expression evaluation routine* col.23:32-35).

Claim 12

The rejection of base claim 11 is incorporated. *Linden* further teaches comprising: said compiler further includes means for compiling said program without generating a specialized path, wherein, when said state of said parameter to be fixed can not be determined, said means for determining the state of said parameter of said predetermined command outputs, as compiled results, said program in said machine language code, which is generated by said means for compiling said program without generating said specialized path, that does not include said specialized path (see at least *instruction sequence, result, constant* paragraph [0038]).

Claims 13-16

Claims recite a computer medium containing a support program controlling a computer for performing the method, which have been addressed in claims 1-2, therefore, are rejected for the same reasons cited in claims 1-2.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> *(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.*

7. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Linden* in view of Shaylor (US 6760907 B2, *Shaylor*).

Claim 5

Linden teaches a program optimization method (see at least *optimizing 14* FIG.2 & associated text; *14, 16* FIG.3 & associated text) for translating, into machine code, the source code for a program written in a programming language (see at least *source instructions, target instructions executable* paragraph [0003]).

Linden does not expressly disclose said programming language is an object-oriented programming language and said optimizing includes detecting one command dynamically during execution, of the commands in said program, for which a method call destination can be identified, and for which the processing speed can be increased by identifying said method call destination; and dynamically generating a path wherefor said method call destination for said detected command is limited in order to increase the processing speed of said command.

However, *Shaylor* discloses said programming language is an object-oriented programming language (see at least *Java source code 201* FIG.2 & associated text) and said optimizing includes detecting one command dynamically during execution (see at least *dynamic compiler 208* col.5:30-col.6:60), of the commands in said program, for which a method call destination can be identified (see at least *"method call", "inlining"* col.2:5-56), and for which the processing speed can be increased by identifying said method call destination (see at least *"method call", "inlining"* col.2:5-56; *optimization of native code, inlining techniques, method calls* col.4:5-42); and dynamically generating a path wherefor said method call destination for said detected command is limited in order to increase the processing speed of said command (see at least *method calls, types of optimizations, inlining* col.6:50-62). *Linden* and *Shaylor* are analogous art because they are directed to dynamic compiler and optimization of executable code. It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the teaching of *Shaylor* into that of *Linden* for the inclusion of object-oriented programming language and detecting and limiting method call destination. And the motivation for doing so would have been to enable optimization of execution speed for platform-independent programs (i.e., source code programs written in object-oriented programming language, such as Java source code) (see at least *Shaylor* col.1:20-45; col.2:48-56).

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571-272-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CP
July 17, 2006

TUAN DAM
SUPERVISORY PATENT EXAMINER